

University of Groningen

A Systematic and Numerically Efficient Procedure for Stable Dynamic Model Inversion of LTI Systems

George, K.; Verhaegen, M.; Scherpen, J.M.A.

Published in:

Proceedings of the 38th IEEE Conference on Decision and Control, 1999

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

1999

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

George, K., Verhaegen, M., & Scherpen, J. M. A. (1999). A Systematic and Numerically Efficient Procedure for Stable Dynamic Model Inversion of LTI Systems. In *Proceedings of the 38th IEEE Conference on Decision and Control, 1999* (Vol. 2, pp. 1881-1886). University of Groningen, Research Institute of Technology and Management.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

A Systematic and Numerically Efficient Procedure for Stable Dynamic Model Inversion of LTI Systems

K. George[†], M. Verhaegen[†] and J.M.A. Scherpen[†]

[†]Control Laboratory, Faculty of Information Technology and Systems,
Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands¹.

[‡]Measurement and Systems Technology Laboratory, Faculty of Applied Physics,
University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands².

Abstract

Output tracking via the novel Stable Dynamic model Inversion (SDI) technique, applicable to non-minimum phase systems, and which naturally takes into account the presence of noise in target time histories, is considered here. We are motivated by the typical need to replicate time signals in the automobile industry. The earlier approaches to stable inversion do not satisfactorily take into account the measurement and system noise, and the zeros of the system are restricted to certain regions in the complex plane.

1 Introduction

Precision output tracking, one of the fundamental problems for control engineers, poses increasingly stringent performance requirements to be satisfied in a variety of applications, notably in the robotics and aerospace industries. In the context of linear systems, it is well-known that perfect tracking is relatively easy to achieve in minimum phase systems. However, output tracking for non-minimum phase systems remains a challenging problem due to the fundamental limitations on transient tracking performance [1] characterised by the number and location of the zeros that are non-minimum phase.

We deal with the problem of output tracking wherein the desired signal is obtained through a data acquisition system, and hence corrupted by noise. We are motivated by the need for time waveform replication (i.e., an accurate reproduction of real or synthesised target time histories) in the automobile industry. Thus complex vibration environments (such as full car endurance tests) may reasonably be recreated in a test laboratory on test-rigs by simulating field measurements thereby saving precious resources. Other applications include acoustical tests of, for instance, automobile components, and driving comfort assessment. Hence there is a need for a technique that obtains a stable inversion of systems that naturally takes into account noise in target time histories. Moreover, discretisation of continuous time systems using a sample and hold lead to non-minimum phase

zeros if the relative degree is greater than two [8], and zeros at unity. Thus, the inversion procedure must account for such zeros as well.

Asymptotic tracking problem in linear systems is achievable [2] if, and only if, a set of linear matrix equations is solvable. This was later generalised [3] to nonlinear system by replacing the linear matrix equations with a set of first order partial differential equations. These approaches asymptotically track any member in a given family of signals generated by an exosystem. The stable inversion approach via a dichotomic split of the system equations of a non-minimum phase plant was introduced [4] to avoid the use of exosystems, and, in the case of non-minimum phase systems, mitigate the poor transient performance by using pre-actuation. Such an approach is related to the classical Hirschorn inverse [5] in the case of minimum phase systems. This procedure, however, does not handle desired signals corrupted by noise, and, cannot handle any zeros on the imaginary axis. The stable inversion approach has been developed in [4] for continuous time systems, and extended to linear time-invariant discrete-time systems in [6].

The problem of trajectory tracking can also be dealt with from the point of view of estimating the state of a system subjected to unknown inputs. Essentially, we first build an 'equivalent system' decoupled from the unknown inputs, and then design a minimum covariance estimator for this equivalent system [7]. This technique was used in [6] to estimate the desired state of an input-decoupled system, and hence compute the desired input. However, it was shown in [6] that this approach is valid only for minimum phase systems. A generalisation to non-minimum phase systems via an inner-outer factorisation of the given system is presented in [6], thereby avoiding the computation of dichotomies. Moreover, it was demonstrated in [6] that under fairly typical conditions, the equivalent system is related to the system matrix of the inverse system [4], and that the states corresponding to the zero dynamics are inadequately reconstructed resulting in inferior output tracking in the presence of noise.

Evidently, there is a need for a different approach that is applicable to non-minimum phase systems, and reasonably

¹{K.George,J.M.A.Scherpen}@its.tudelft.nl

²M.Verhaegen@tn.utwente.nl

accounts for noise on the desired trajectory. In this paper we present the Stable Dynamic model Inversion (SDI) technique that satisfy these objectives. This procedure involves augmenting the observable state space model of the given plant by a model for the input time history, and the input sequence is estimated by a Kalman filter design. We will show that this technique is more general than the earlier schemes in that it is not limited by the presence of zeros in regions of the complex plane, and by a suitable choice of the Kalman gain, the presence of noise in target time histories is easily taken into account.

The basic SDI technique is only restricted by the possible presence of zeros at $z = 1$. We generalise by factoring the given transfer function into $P(z) = P_1(z)P_2(z)$, where the zeros of $P_1(z)$ are located at $z = 1$, including multiplicity. We then pre-process the desired signal by $P_1^{-1}(z)$, and apply the SDI procedure to $P_2(z)$ and this pre-processed desired signal. In the sequel, we provide two algorithms that provide the desired factorisation. The first algorithm is based on the *a priori* knowledge of the presence of the zeros, and $P_1(z)$ is an all-zero function. The second algorithm is based on the zero dislocation procedure presented in [9], and avoids the explicit computation of zeros. Moreover, since the decomposition is obtained using only unitary matrices, it is numerically efficient.

This paper is organised as follows. In Section 2 we present SDI for discrete time systems, and generalised in Section 3. An illustrative benchmark example supporting the contributions of this paper is treated in Section 4.

2 Stable Dynamic Model Inversion (SDI)

We describe an approach that inverts the model of a plant whilst overcoming the main drawbacks of existing techniques, namely, that of handling either non-minimum phase zeros, or zeros on the unit circle, and noise in the desired signal. Consider the following state space realization of a linear time invariant system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Gw_k \\ z_k &= Cx_k + Du_k + v_k \end{aligned} \quad (1)$$

We assume that $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, $y_k \in \mathbb{R}^p$, w_k and v_k are white noise processes with covariances Q_w and R_v respectively, uncorrelated with each other, and with the initial condition x_0 .

Problem 1 *Given the above state space model of a plant, a target time history $\{z_k\}_{k=1}^N$, determine $\{\hat{u}_k\}_{k=0}^N$ such that $\mathcal{E}\{\|z_k - \hat{z}_k\|^2\}$ and $\mathcal{E}\{\|u_k - \hat{u}_k\|^2\}$ are minimised.*

This is illustrated in Fig. 1, where Σ is represented by eqn. (1). Our objective is to generate a desired input sequence \hat{u}_k by obtaining a suitable ‘inverse system’ Σ_{inv} that relates this input sequence to the given measurement sequence z_k .

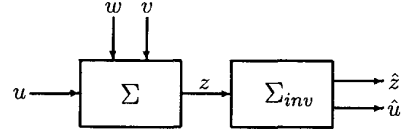


Figure 1: Stable dynamic model inversion.

The SDI technique is based on augmenting the given state space model (1) by a reasonable model for the input sequence and then designing a Kalman filter to provide an estimate of the input sequence from the measurements z_k . From intuitive and physical reasoning, it seems realistic to model the input signal u_k as follows

$$u_{k+1} = u_k + \eta_k \quad (2)$$

for some η_k . For simplicity, we assume that η_k is white noise with covariance Q_η , and uncorrelated with w_k and v_k . The resulting augmented system is as follows:

$$\begin{aligned} x_{a,k+1} &= A_a x_{a,k} + G_a w_{a,k} \\ z_k &= C_a x_{a,k} + H_a w_{a,k} \end{aligned}$$

where $x_{a,k} = (x'_k \ u'_k)'$ and $w_{a,k} = (w'_k \ v'_k \ \eta'_k)'$. The following result readily follows from the definitions of observability and the zeros of a system:

Lemma 2 *Suppose the pair (C, A) is observable. The pair (C_a, A_a) is observable if, and only if, $z = 1$ is not a zero of the system (A, B, C, D) .*

Therefore, under the conditions for observability of the augmented system given in Lemma 2 we can set up a Kalman filter to estimate the input signal:

$$\hat{x}_{a,k+1|k} = A_a \hat{x}_{a,k|k-1} - K (C_a \hat{x}_{a,k|k-1} - z_k)$$

where $K \triangleq \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} \triangleq A_a P C_a' (C_a P C_a' + R_v)^{-1}$ and P satisfies the standard filter Riccati equation. The Kalman gains K_1 and K_2 respectively correspond to the state x_k and the input u_k . Clearly, the transfer function from the measurements z_k to the estimate of the input $\hat{u}_{k|k-1}$ (i.e., the inverse system) is as follows:

$$\Sigma_{inv} = \begin{pmatrix} 0 & I \end{pmatrix} (zI - A_a + K C_a)^{-1} K \quad (3)$$

We note that the poles of Σ_{inv} are the relocated eigenvalues of the system matrix A_a . Moreover, we have the following result on the zeros of the inverse system; its proof is rather straightforward from the definition of zeros:

Lemma 3 *Every eigenvalue of the system matrix A is a zero of the inverse system (3).*

Evidently, the augmented system is observable at all points on the complex plane if the given plant has no zeros at

$z = 1$. Thus, this technique is more general than the earlier schemes in that it does not restrict the presence of zeros to the stable and unstable regions in the strict sense [4, 6], or to the stable region [7]. Moreover, by suitably designing the Kalman filter, we can easily take into account the presence of noise in target time histories. We recall that the method provided in [4] does not handle noise in target time histories, and the states corresponding to the zero dynamics are poorly re-constructed in the technique presented in [7]. In the next section we present algorithms to recover observability if the original system has zeros at unity.

3 Recovering Observability of the Augmented System

In this section, we devise methods to overcome the drawback of the SDI technique presented in the previous section. We recall that the augmented system loses observability if, and only if, there are zeros located at $z = 1$. Thus, in principle, the only offending point is $z = 1$; this zero is dislocated in order to recover observability. In practice, however, for numerical efficiency, it is recommended that any zeros located within a circle of radius ϵ centred at unity be dislocated. We present two methods for dislocating the zeros in this region and grouping them together into one function.

Given the following system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \quad (4)$$

we aim to obtain the following decomposition:

$$\begin{aligned} P(z) &= C(zI - A)^{-1}B + D \\ &= P_1(z)P_2(z) \end{aligned} \quad (5)$$

where the zeros of $P_1(z)$ are the zeros of $P(z)$ located within a circle of radius ϵ centred at unity. Thus,

$$\begin{aligned} y &= P_1 P_2 u \\ \Rightarrow P_1^{-1} y &= P_2 u \end{aligned}$$

We first compute $\tilde{y} \triangleq P_1^{-1} y$ and then apply the SDI procedure to the following system:

$$\tilde{y} = P_2 u \quad (6)$$

We note that \tilde{y} should have a mixed causal and anti-causal structure if $P_1^{-1}(z)$ is unstable; however, it has been our experience, that marginal instability does not cause serious problems during the computation of \tilde{y} .

3.1 Extraction of an All-Pole $P_1^{-1}(z)$

This technique assumes an *a priori* knowledge of the exact location of the zeros and makes $P_1^{-1}(z)$ an all-pole transfer function. By observing the Markov parameters of a state space model of $P(z)$, the following result gives the state space model of $P_2(z)$:

Lemma 4 Let $P(z) = C(zI - A)^{-1}B + D$ be any transfer matrix.

1. If $P(z)$ has a zero at $-\lambda$, then

$$P(z) = (z + \lambda)C_n(zI - A)^{-1}B$$

$$\text{where } C_n = \begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} A + \lambda I & B \end{pmatrix}^\dagger.$$

2. If $P(z)$ has a pair of zeros represented by the roots of the polynomial $z^2 + az + b$ then

$$P(z) = (z^2 + az + b)C_n(zI - A)^{-1}B$$

$$\text{where } C_n = \begin{pmatrix} C & D \end{pmatrix} \begin{pmatrix} A^2 + aA + bI & AB \end{pmatrix}^\dagger.$$

Here X^\dagger represents the Moore-Penrose inverse of the matrix X . Lemma 4 is a more general result, and we recursively apply this to dislocate any zeros in the region of the complex plane defined by a circle of radius ϵ centred at unity.

3.2 Direct Extraction of State Space Model of $P_1^{-1}(z)$

We now describe a method adapted from the more general method of dislocating zeros discussed in [9]. This algorithm avoids the explicit computation of zeros, and the dislocation is obtained using only unitary matrices and hence is numerically efficient. Essentially, this approach first reduces a matrix pencil associated with (4) into a generalised Schur form in which the zeros of the plant (4) is separated into those that are associated with a region in the complex plane Γ , and those that are associated with Γ^c , its complement. Then, the minimal realization of $P^{-1}(z)$ is directly obtained via a solution of a pole placement problem. Therefore, we seek a factorisation (5) [9] such that:

1. $P_1(z)$ is a $p \times p$ nonsingular transfer matrix.
2. Given a region Γ of the complex plane, the zeros of $P_1^{-1}(z)$ and $P_2(z)$ lie in Γ .
3. The McMillan degree of $P_1(z)$ is minimal and exactly equal to the number of zeros of $P(z)$ in Γ^c .

In this paper, we denote the region enclosed by a circle of radius ϵ centred at $z = 1$ by Γ^c .

The numerically efficient method of performing this factorisation is summarised by the following algorithm [9]. (The notation is similar to that in [9], and hence, for brevity, not explained in the sequel.)

Algorithm 1 1. Find unitary matrices Q and U to decompose as follows:

$$\begin{aligned} &\left(\begin{array}{c|c} Q^* (\lambda I_n - A) Q & Q^* B \\ \hline -C Q & D \end{array} \right) U \\ &= \left(\begin{array}{ccc|c} \lambda I_{n_1} - A_{11} & -A_{12} & -A_{13} & B_1 \\ -A_{21} & \lambda I_{n_2} - A_{22} & -A_{23} & B_2 \\ -A_{31} & -A_{32} & \lambda I_{n_3} - A_{33} & B_3 \\ \hline -C_1 & -C_2 & -C_3 & D \end{array} \right) U \\ &= \left(\begin{array}{ccc|c} \lambda \tilde{E}_{11} - \tilde{A}_{11} & \lambda \tilde{E}_{12} - \tilde{A}_{12} & \lambda \tilde{E}_{13} - \tilde{A}_{13} & \lambda \tilde{F}_1 - \tilde{B}_1 \\ 0 & \lambda \tilde{E}_{22} - \tilde{A}_{22} & \lambda \tilde{E}_{23} - \tilde{A}_{23} & \lambda \tilde{F}_2 - \tilde{B}_2 \\ 0 & 0 & \lambda \tilde{E}_{33} - \tilde{A}_{33} & \lambda \tilde{F}_3 - \tilde{B}_3 \\ \hline 0 & 0 & 0 & \tilde{D} \end{array} \right) \end{aligned}$$

where \tilde{D} has linearly independent columns.

2. Define $\hat{F} = \hat{A}_{22}\hat{E}_{22}^{-1}$.

3. Solve the following equation for Y_3 and \hat{Y}_3 :

$$Y_3(\lambda\hat{E}_{33} - \hat{A}_{33}) - (\lambda I - \hat{F})\hat{Y}_3 = -(\lambda\hat{E}_{23} - \hat{A}_{23}) \quad (7)$$

4. Define $\hat{Y}_4 = \hat{F}_2 + Y_3\hat{F}_3$, and solve for \hat{G} from

$$\hat{G}\hat{D} = \hat{B}_2 + Y_3\hat{B}_3 - \hat{F}\hat{Y}_4$$

5. Choose K such that the eigenvalues of $\hat{F} + \hat{G}K$ lie in Γ . Let

$$\begin{pmatrix} \hat{H} & J \end{pmatrix} = M \begin{pmatrix} -K & I \end{pmatrix}$$

for some arbitrary invertible M . (This implies that \hat{H} and J are chosen such that the zeros of $\hat{H}(zI - \hat{F})^{-1}\hat{G} + J$ lie in Γ .)

6. Finally,

$$P_1^{-1}(z) \longleftrightarrow \left(\begin{array}{c|c} X_2^{-1}\hat{F}X_2 & X_2^{-1}\hat{G} \\ \hline \hat{H}X_2 & J \end{array} \right)$$

$$P_2(z) \longleftrightarrow \left(\begin{array}{c|c} A & B \\ \hline JC + HX & JD \end{array} \right)$$

We note that the solution of eqn. (7) was not explicitly considered in [9]. In order to adapt the solution of generalised Sylvester equations given in [10], we note that Y_3 and \hat{Y}_3 is the solution for eqn. (7) if, and only if, they solve the following equation:

$$\begin{pmatrix} I & -Y_3 \\ 0 & I \end{pmatrix} (M - \lambda N) \begin{pmatrix} I & \hat{Y}_3 \\ 0 & I \end{pmatrix} = \begin{pmatrix} \hat{F} & 0 \\ 0 & \hat{A}_{33} \end{pmatrix} - \lambda \begin{pmatrix} I & 0 \\ 0 & \hat{E}_{33} \end{pmatrix} \quad (8)$$

where

$$M = \begin{pmatrix} \hat{F} & -\hat{A}_{23} \\ 0 & \hat{A}_{33} \end{pmatrix} \quad N = \begin{pmatrix} I & -\hat{E}_{23} \\ 0 & \hat{E}_{33} \end{pmatrix}$$

The first step in the computation of the solution to the generalised Sylvester equation is to obtain orthogonal matrices Q and Z such that $Q'(M - \lambda N)Z = (\hat{M} - \lambda\hat{N})$ where \hat{M} is in upper Hessenberg form and \hat{N} is upper triangular. (Observe that \hat{A}_{33} is already in upper Hessenberg form, and N in upper triangular form.) We will assume that M and N are in this required form. Let \hat{F} consist of p^2 blocks \hat{F}_{ij} , where each of the p diagonal blocks is either 1×1 or 2×2 . Similarly, let \hat{A}_{33} consist of q^2 blocks $\hat{A}_{33,ij}$, with the diagonal blocks either 1×1 or 2×2 . The other matrices are partitioned consistently with \hat{F} and \hat{A}_{33} . The solution is then obtained as follows [10]: For $j = 1, 2, \dots, q$, $i = p, p-1, \dots, 1$, we first solve the subsystem

$$\begin{aligned} \hat{F}_{ii}\hat{Y}_{3,ij} - Y_{3,ij}\hat{A}_{33,jj} &= \hat{A}_{23,ij} \\ \hat{Y}_{3,ij} - Y_{3,ij}\hat{E}_{33,jj} &= \hat{E}_{23,ij} \end{aligned} \quad (9)$$

and then substitute the solution $\hat{Y}_{3,ij}$ and $Y_{3,ij}$ into the remaining equations. Finally, we note that the solution to subsystem (9) is obtained after rewriting the equations using Kronecker products.

We have presented in this section two methods that generalise the SDI procedure presented in the previous section. Thus, if the given plant has zeros at unity, or in its vicinity, we are now able to dislocate them in a reliable and numerical efficient manner. We note that we can apply the results of Lemma 4 immediately after the third step in Algorithm 1 by computing the eigenvalues of \hat{F} (i.e., the zeros of (4) in Γ^c). However, this is not recommended in practice especially when there are clusters of zeros in Γ^c . We also remark that the SDI procedure of Section 2, and its generalisation in this section can easily be extended to linear time invariant continuous time systems. For such systems the Γ^c represents a circle of radius ϵ centred at the origin. In the next section, we validate the results of this paper on a benchmark example.

4 Benchmark Example: A Quarter Car

The quarter-car model is the simplest model that includes a proper representation of the problem of controlling wheel load variations. It contains no representation of geometric effects of having four wheels and offers no possibility of studying longitudinal interactions or the use of front suspension state information to improve the performance at the rear. Moreover, it cannot describe problems related to handling. However, it does appear to contain the most basic features of the real problem and gives rise to design thinking which accords with experience. The quarter car can be expressed as a two-mass model shown in Fig. 2, and governed by the following differential equations:

$$\begin{aligned} m_c \ddot{x}_c &= -c_c(x_c - x_w) - d_c(\dot{x}_c - \dot{x}_w) \\ m_w \ddot{x}_w &= -c_w(x_w - x_b) + c_c(x_c - x_w) + d_c(\dot{x}_c - \dot{x}_w) \end{aligned}$$

Here, the subscripts c and w respectively denote the body and the wheel of the car, x denotes the displacement, m the mass, c the stiffness, and d the damping coefficient. The road displacement input is denoted by x_b . For the analysis of, say, the fatigue of a car, we wish to replicate the acceleration data on the car body, which has been obtained in a field test. This is done on a test rig by the control of shaker displacement. The system input u is now the shaker displacement instead of the road displacement x_b . The system output is the acceleration of the sprung (or, body) mass, \ddot{x}_c .

In our scheme, the controller is discrete. A DAC is needed to obtain the necessary control signal to drive the plant. With a sampling rate of 200 Hz and a zero order hold, the discretised transfer function (obtained through MATLAB) of

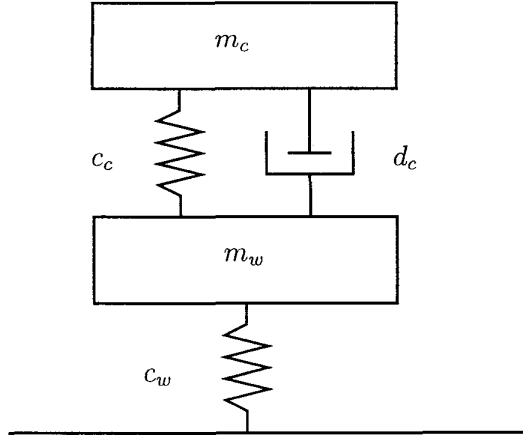


Figure 2: Quarter Car

the quarter car is as given below:

$$P(z) = \frac{162.64(z^3 - 2.9631z^2 + 2.9262z - 0.9631)}{z^4 - 3.6652z^3 + 5.1437z^2 - 3.2875z + 0.80911} \quad (10)$$

The zeros are at 1.0, 1.0004, and 0.9627, and hence the plant is marginally non-minimum phase, due to sampling. (We note that the continuous time model has one stable zero and two zeros at the origin.) We choose the region Γ^c with $\epsilon = 0.001$, thereby factoring out the zeros at 1.0 and 1.0004.

The desired signal y_d given in Fig. 3(a) represents the acceleration of the sprung mass caused by a typical road profile. A corrupted version of this signal y_{dc} given in Fig. 3(b) represents the measured acceleration of the sprung mass. We use the following measure Variance Accounted For (VAF) to compare two signals r_1 and r_2 :

$$\text{VAF}(r_1, r_2) = \left(1 - \frac{\text{variance}(r_1 - r_2)}{\text{variance}(r_1)}\right) 100\%$$

The value indicates how close one signal is to the other; if they are completely equal, VAF is 100%. For the desired signal y_d and its corrupted version y_{dc} we have $\text{VAF}(y_d, y_{dc}) = 63.98\%$, which represents a fairly worse-case condition.

Assuming an *a priori* knowledge of their locations, we recursively dislocate the zeros using Lemma 4 to yield the following final factorisation (5):

$$P_1^{-1} = \frac{1}{z^2 - 2.0004z + 1.0004}$$

$$P_2 = \frac{162.64(z - 0.9627)}{z^4 - 3.6652z^3 + 5.1437z^2 - 3.2875z + 0.80911}$$

The poles of $P_1^{-1}(z)$ are at 1.0 and 1.0004; the zero of $P_2(z)$ is at 0.9627. We remark on the choice of covariances Q_w , Q_η and R_v . For the noise-free case, we assign $Q_w = 0$, $Q_\eta = I$ and $R_v = I$. For the other case, the choice of R_v

clearly depends on the noise level in the target time history. Since, for the example considered in this section, the corrupted desired signal y_{dc} represents a worse-case condition, a high value of R_v is required, and is of the order of 10^7 . We denote the output of the given plant (10) as y_o when the input to the inverse system is y_d ; correspondingly, y_{oc} represents the output of the given plant when the input to the system is y_{dc} (refer Fig. 1); these are as shown in Fig. 4. Note that both signals y_d and y_o perfectly superimpose in Fig. 4(a). As expected from the response plots, for the noise-free case, we have $\text{VAF}(y_d, y_o) = 100\%$, which is a rather good performance. For the other case, we have $\text{VAF}(y_d, y_{oc}) = 89.32\%$, which is a rather reasonable performance considering the fact that y_{dc} represents y_d rather poorly.

We now apply the state space approach to dislocate the zeros given in Algorithm 1 to obtain the following:

$$P_1^{-1} = \frac{z^2 + 0.03z + 0.0002}{z^2 - 2.0004z + 1.0004}$$

$$P_2 = \frac{162.64(z^3 - 0.9327z^2 - 0.028682z - 0.00019254)}{z^4 - 3.6652z^3 + 5.1437z^2 - 3.2875z + 0.80911}$$

The poles of $P_1^{-1}(z)$ are at 1.0 and 1.0004; its zeros have been placed at -0.02 and -0.01 yielding an invertible transfer function. The zeros of $P_2(z)$ are at 0.9627, -0.02 and -0.01 . The outputs y_o and y_{oc} are as shown in Fig. 5. Again, both signals y_d and y_o perfectly superimpose in Fig. 5(a). Thus, we have $\text{VAF}(y_d, y_o) = 99.99\%$; for the noisy case, $\text{VAF}(y_d, y_{oc}) = 89.25\%$.

5 Conclusions

The earlier techniques for the stable inversion of systems are inadequate in the sense that the zeros of the systems are restricted to certain regions of the complex plane, and exhibits a not-so-satisfactory performance in the presence of noise in target time histories. The Stable Dynamic model Inversion (SDI) technique overcomes these main drawbacks. The basic SDI procedure is applicable to any plant that does not have zeros at unity. This is generalised to include all plants by a numerically efficient technique for the dislocation of such zeros. Moreover, by a suitable design of the Kalman Filter, the technique naturally tries to compensate for noise in target time histories. A MATLAB toolbox for SDI has successfully been used in several applications.

Acknowledgements

This work was supported by Brite-Euram under contract number BRPR-CT97-0508 and project number BE-97-4186

References

- [1] L. Qiu and E. J. Davison, "Performance limitations of non-minimum phase systems in the servomechanism problem," *Automatica*, vol. 29, no. 2, pp. 337-349, 1993.

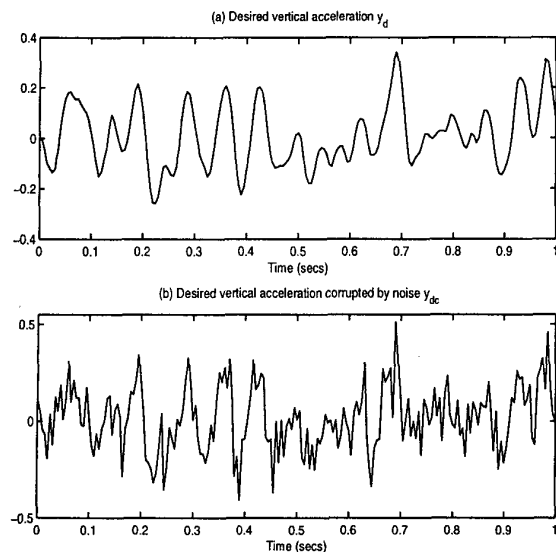


Figure 3: The desired and corrupted vertical acceleration.

- [2] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, pp. 457–465, 1976.
- [3] A. Isidori and C. I. Byrnes, "Output regulation of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 2, pp. 131–140, February 1990.
- [4] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Transactions on Automatic Control*, vol. 41, no. 7, pp. 930–942, July 1996.
- [5] R. M. Hirschorn, "Invertibility of nonlinear control systems," *SIAM Journal of Control and Optimization*, vol. 17, no. 2, pp. 289–297, March 1979.
- [6] K. George, M. Verhaegen, and J. M. A. Scherpen, "Stable inversion of MIMO linear discrete time non-minimum phase systems," in *Proceedings of the 7th Mediterranean Conference on Control and Automation*, Haifa, Israel, June 1999, pp. 267–281.
- [7] M. Hou and R. J. Patton, "Optimal filtering for systems with unknown inputs," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 445–449, March 1998.
- [8] K. J. Åström, P. Hagander, and J. Sternby, "Zeros of sampled systems," *Automatica*, vol. 20, no. 1, pp. 31–38, 1984.
- [9] P. Van Dooren, "Rational and polynomial matrix factorizations via recursive pole-zero cancellation," *Linear Algebra and its Applications*, vol. 137/138, pp. 663–697, 1990.
- [10] B. Kågström and L. Westin, "Generalized Schur methods with condition estimators for solving the generalized Sylvester equation," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 745–751, July 1989.

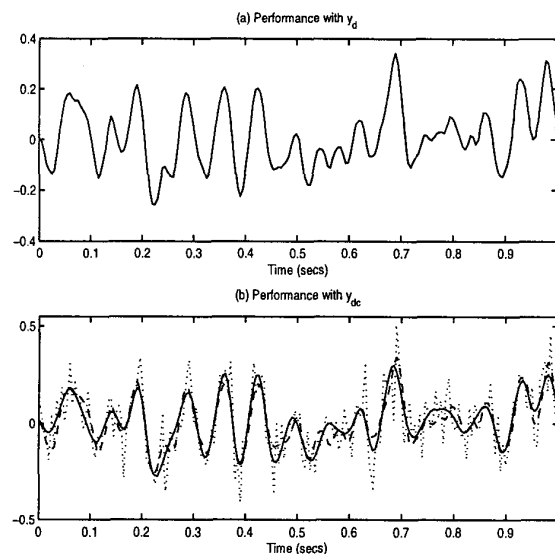


Figure 4: Performance of first method for dislocation of zeros. (a) With y_d : — Output of system; -- Desired Signal. The two signals perfectly superimpose. (b) With y_{dc} : — Output of system; -- Desired signal y_d ; ... Corrupted desired signal y_{dc} .

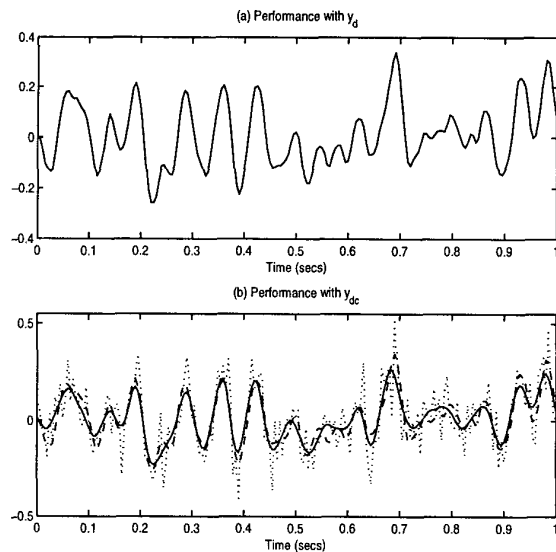


Figure 5: Performance of second method for dislocation of zeros. (a) With y_d : — Output of system; -- Desired Signal. The two signals perfectly superimpose. (b) With y_{dc} : — Output of system; -- Desired signal y_d ; ... Corrupted desired signal y_{dc} .